
Subject: Re: Spawning EML Scripts in IDL
Posted by [Jean H.](#) on Wed, 13 Dec 2006 20:32:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

> Is
> there a way to have IDL check if the file in question is being used by
> another application? I'd like to have IDL report a message stating if
> this is the case, and for IDL to move on to the next file without
> stopping the script. Does anyone have any suggestions?
>
> Matt
>

You may want to play with FILE_TEST and the /read and /write keywords

Jean

Subject: Re: Spawning EML Scripts in IDL
Posted by [Robbie](#) on Thu, 14 Dec 2006 00:01:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

File locking is generally operating system dependent. Although, the file lock should be released if you terminate the EML process. You can find the process id by using the PID keyword in spawn. You can then terminate it using kill (UNIX) or a call to a MFC function (Windows)

Robbie

Subject: Re: Spawning EML Scripts in IDL
Posted by [Rick Towler](#) on Thu, 14 Dec 2006 00:13:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

On windows you can use the WMI (Windows Management Interface). You could probably do this via IDLcomIDispatch, but spawning vb script is easier. Something like this should suffice:

```
Dim objWMIService, objProcess, colProcess
Dim strComputer, strProcessKill
strComputer = "."
strProcessKill = "eml.exe"
```

```
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" _
& strComputer & "\root\cimv2")
```

```
Set colProcess = objWMIService.ExecQuery _
("Select * from Win32_Process Where Name = " & strProcessKill )
For Each objProcess in colProcess
objProcess.Terminate()
Next
```

WScript.Quit

-Rick

Robbie wrote:

> File locking is generally operating system dependent. Although, the
> file lock should be released if you terminate the EML process. You can
> find the process id by using the PID keyword in spawn. You can then
> terminate it using kill (UNIX) or a call to a MFC function (Windows)
>
> Robbie
>

Subject: Re: Spawning EML Scripts in IDL
Posted by [Matt\[1\]](#) on Thu, 14 Dec 2006 16:38:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the great replies:

To recap, it seems like there are two steps that I need to get a grasp of (followed by what I am beginning to understand from responses - bear with me, I'm a total newbie) -

1. Identify whether the file I want to use in IDL is being used by another program.
2. If it is in fact being used, identify the process using it and kill it

With regard to 1. Unfortunately, I can't use the SPAWN PID keyword to identify the process because I'm actually spawning a windows BAT file that in turn spawns the EML script. So the PID will identify the BAT process, which finishes in less than one second. Also, the File_test read/write keywords, from what I can tell, identify whether you have read/write permissions within that directory (am I wrong? In any case, the keywords are telling me that I can read/write a file that is being used by another process). SO... I'm beginning to think that I need to get a better grasp of IDL error handling functions. Can I just have IDL report/catch/capture the error (when it tries to move a file that it can't) and move on to the next line without going back to interactive mode?

With regard to 2. It seems that killing an app can't be done through IDL but could be done through vb. Rick, thanks for the info, and thanks for giving me something to do over the weekend because I don't know ANY vb...

Thanks helpful people...

Matt

Rick Towler wrote:

> On windows you can use the WMI (Windows Management Interface). You
> could probably do this via IDLcomIDispatch, but spawning vb script is
> easier. Something like this should suffice:

```
>  
> Dim objWMIService, objProcess, colProcess  
> Dim strComputer, strProcessKill  
> strComputer = "."  
> strProcessKill = "eml.exe"  
>  
> Set objWMIService = GetObject("winmgmts:" _  
> & "{impersonationLevel=impersonate}!\\" _  
> & strComputer & "\root\cimv2")  
>  
> Set colProcess = objWMIService.ExecQuery _  
> ("Select * from Win32_Process Where Name = " & strProcessKill )  
> For Each objProcess in colProcess  
> objProcess.Terminate()  
> Next  
>  
> WScript.Quit  
>  
> -Rick
```

> Robbie wrote:

```
>> File locking is generally operating system dependent. Although, the  
>> file lock should be released if you terminate the EML process. You can  
>> find the process id by using the PID keyword in spawn. You can then  
>> terminate it using kill (UNIX) or a call to a MFC function (Windows)  
>>  
>> Robbie  
>>
```

Subject: Re: Spawning EML Scripts in IDL
Posted by [Rick Towler](#) on Thu, 14 Dec 2006 17:54:54 GMT

Matt wrote:

> Thanks for the great replies:
>
> To recap, it seems like there are two steps that I need to get a grasp
> of (followed by what I am beginning to understand from responses - bear
> with me, I'm a total newbie) -
>
> 1. Identify whether the file I want to use in IDL is being used by
> another program.
>
> 2. If it is in fact being used, identify the process using it and kill
> it
>
> With regard to 1. Unfortunately, I can't use the SPAWN PID keyword to
> identify the process because I'm actually spawning a windows BAT file
> that in turn spawns the EML script. So the PID will identify the BAT
> process, which finishes in less than one second. Also, the File_test
> read/write keywords, from what I can tell, identify whether you have
> read/write permissions within that directory (am I wrong? In any case,
> the keywords are telling me that I can read/write a file that is being
> used by another process). SO... I'm beginning to think that I need to
> get a better grasp of IDL error handling functions. Can I just have IDL
> report/catch/capture the error (when it tries to move a file that it
> can't) and move on to the next line without going back to interactive
> mode?

You should be able to "catch" the error. Something like:

```
n = 0
while n < nFiles

    catch, error
    if (error ne 0) then begin
        catch, /cancel
        spawn, 'c:\kill_ems.vbs'
        file_move, sourcePath+outfile, errorPath+outfile
    endif

    ; rest of script here
    n = n + 1

endwhile
```

When your script runs into a problem moving a locked file it will jump into the catch block, kill the process, move the file someplace, and then continue out the bottom.

- > With regard to 2. It seems that killing an app can't be done through
- > IDL but could be done through vb. Rick, thanks for the info, and thanks
- > for giving me something to do over the weekend because I don't know ANY
- > vb...

You should be able to cut and paste that into a text editor, save it as "kill_eml.vbs" and you're set. Of course you will need edit the strProcessKill variable to properly reflect the name of your EML process. The only issue you may run into is if you are running multiple EML scripts simultaneously. My simple script kills processes by name, and it kills all of what it finds. If you need to kill particular processes, you will need to write a vbs script that spawns the EML process and returns the pid. Your kill script would then accept the PID as an argument and kill based on PID, not name. But from your initial description, this shouldn't be a problem.

-Rick

- > Thanks helpful people...
- >
- > Matt
- >
- >
- > Rick Towler wrote:
- >> On windows you can use the WMI (Windows Management Interface). You
- >> could probably do this via IDLcomIDispatch, but spawning vb script is
- >> easier. Something like this should suffice:
- >>
- >> Dim objWMIService, objProcess, colProcess
- >> Dim strComputer, strProcessKill
- >> strComputer = "."
- >> strProcessKill = "eml.exe"
- >>
- >> Set objWMIService = GetObject("winmgmts:" _
- >> & "{impersonationLevel=impersonate}!\\" _
- >> & strComputer & "\root\cimv2")
- >>
- >> Set colProcess = objWMIService.ExecQuery _
- >> ("Select * from Win32_Process Where Name = " & strProcessKill)
- >> For Each objProcess in colProcess
- >> objProcess.Terminate()
- >> Next
- >>
- >> WScript.Quit

```
>>
>> -Rick
>>
>> Robbie wrote:
>>> File locking is generally operating system dependent. Although, the
>>> file lock should be released if you terminate the EML process. You can
>>> find the process id by using the PID keyword in spawn. You can then
>>> terminate it using kill (UNIX) or a call to a MFC function (Windows)
>>>
>>> Robbie
>>>
>
```
